In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
from sklearn.model_selection import train_test_split
from sklearn.linear_model import PassiveAggressiveRegressor
```

In [2]:

```
df = pd.read_csv(r"C:\Users\shree\Downloads\archive\Instagram data.csv", encoding = 'latin1')
```

In [3]:

```
# checking for missing values
```

In [4]:

```
df.isnull().sum()
```

Out[4]:

```
Impressions       0
From Home         0
From Hashtags     0
From Explore      0
From Other        0
Saves             0
Comments          0
Shares            0
Likes             0
Profile Visits    0
Follows           0
Caption           0
Hashtags          0
dtype: int64
```

In [5]:

```
df.info()
```

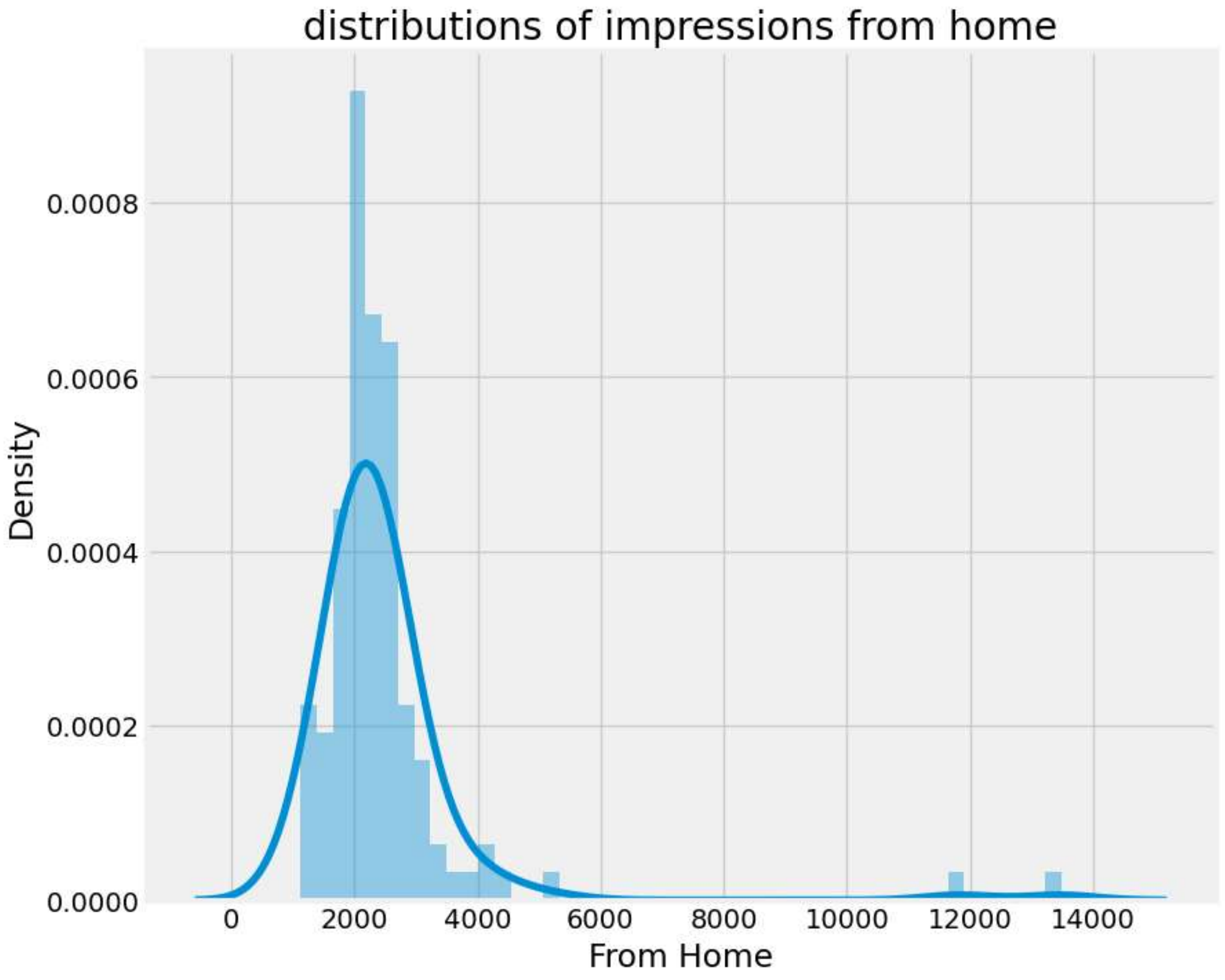
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 119 entries, 0 to 118
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   Impressions     119 non-null    int64 
 1   From Home       119 non-null    int64 
 2   From Hashtags   119 non-null    int64 
 3   From Explore    119 non-null    int64 
 4   From Other      119 non-null    int64 
 5   Saves           119 non-null    int64 
 6   Comments        119 non-null    int64 
 7   Shares          119 non-null    int64 
 8   Likes           119 non-null    int64 
 9   Profile Visits  119 non-null    int64 
 10  Follows         119 non-null    int64 
 11  Caption         119 non-null    object
 12  Hashtags        119 non-null    object
dtypes: int64(11), object(2)
memory usage: 12.2+ KB
```

analyzing instagram reach

In [6]:

```
plt.figure(figsize =( 10,8))
plt.style.use("fivethirtyeight")
plt.title("distributions of impressions from home")
sns.distplot(df["From Home"])
plt.show()
```

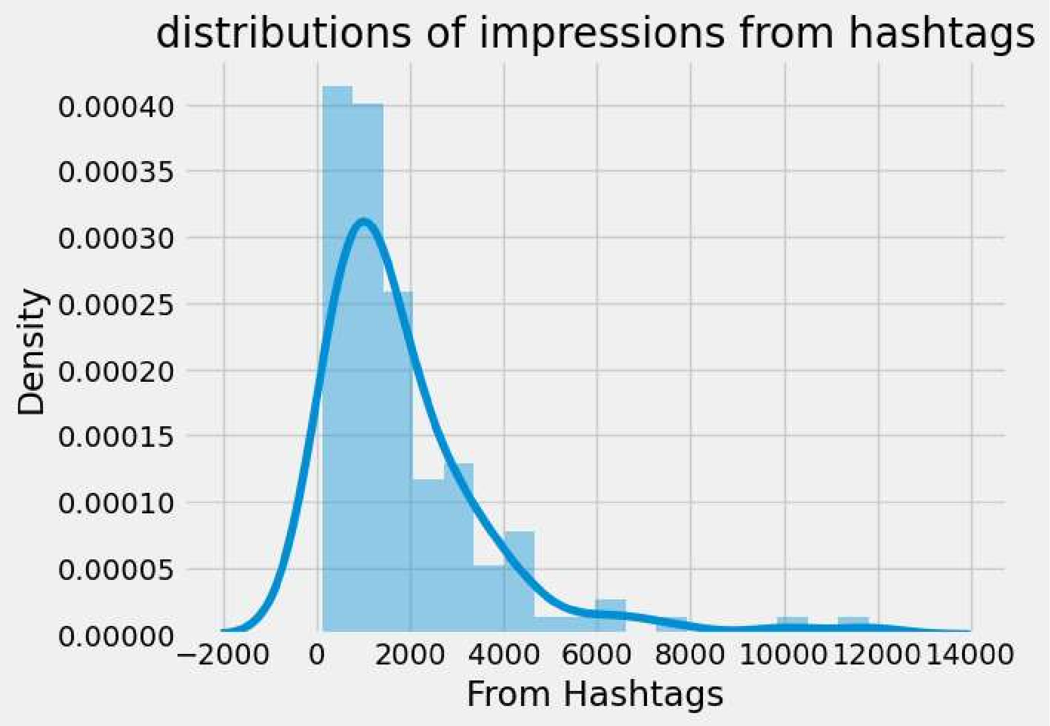
```
C:\Users\shree\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot`
(a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



In [7]:

```
plt.title("distributions of impressions from hashtags")
sns.distplot(df["From Hashtags"])
plt.show()
```

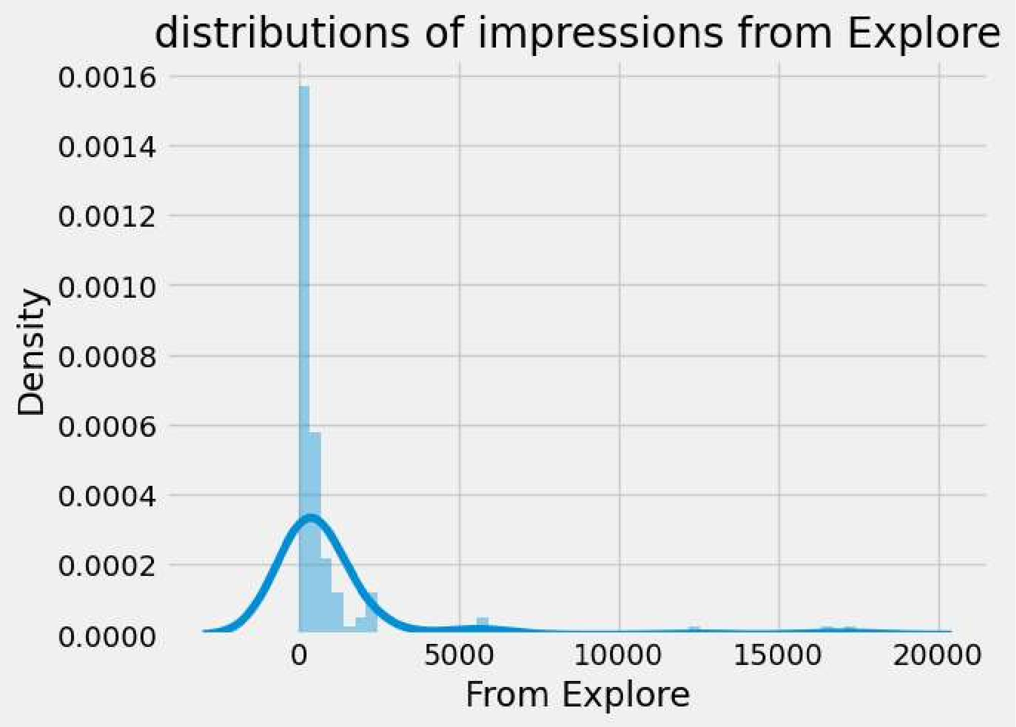
```
C:\Users\shree\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot`
(a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



In [8]:

```
plt.title("distributions of impressions from Explore")
sns.distplot(df["From Explore"])
plt.show()
```

```
C:\Users\shree\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a d
eprecated function and will be removed in a future version. Please adapt your code to use either `displot`
(a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



In [9]:

```
# The explore section of Instagram is the recommendation system of Instagram.
#It recommends posts to the users based on their preferences and interests.
#By looking at the impressions I have received from the explore section, I can say that Instagram does not recommend ou
#Some posts have received a good reach from the explore section, but it’s still very low compared to the reach I receiv
```

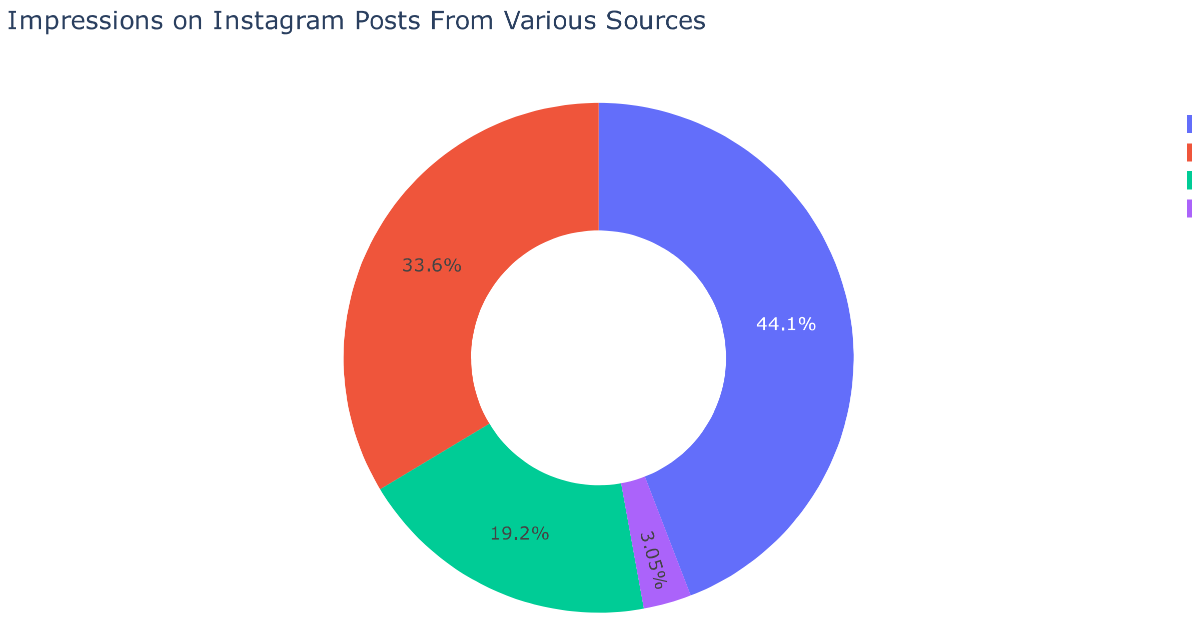
percentage of impression from various sources on instagram

In [10]:

```
home = df["From Home"].sum()
hashtags = df["From Hashtags"].sum()
explore = df["From Explore"].sum()
other = df["From Other"].sum()

labels = ['From Home','From Hashtags','From Explore','Other']
values = [home, hashtags, explore, other]

fig = px.pie(df, values=values, names=labels,
             title='Impressions on Instagram Posts From Various Sources', hole=0.5)
fig.show()
```



Impressions on Instagram Posts From Various Sources

In [11]:

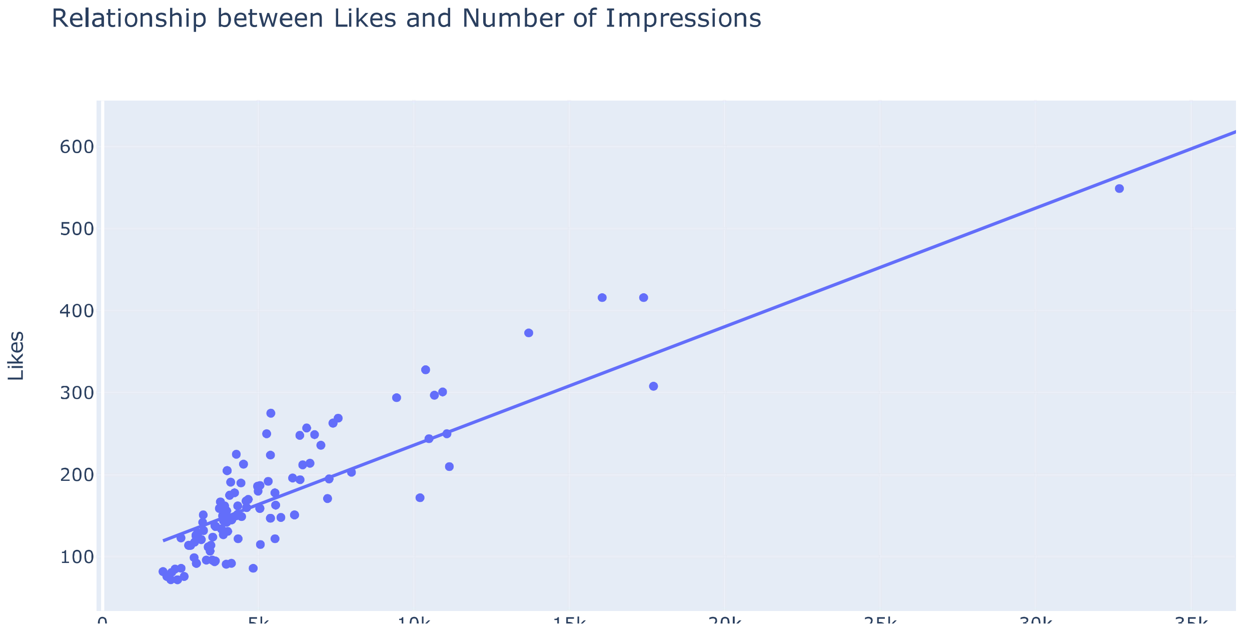
```
# so from the above donut 44.1% is from my followers , 33.6% is from hashtags ,19.2% is from explore and 3.05% is from
```

analyzing relationships

lets have a look at the relationship between number of likes and the number of impressions in instagram posts

In [12]:

```
figure = px.scatter( data_frame = df, x = "Impressions" , y = "Likes" ,trendline = "ols",
                    title = "Relationship between Likes and Number of Impressions" )
figure.show()
```



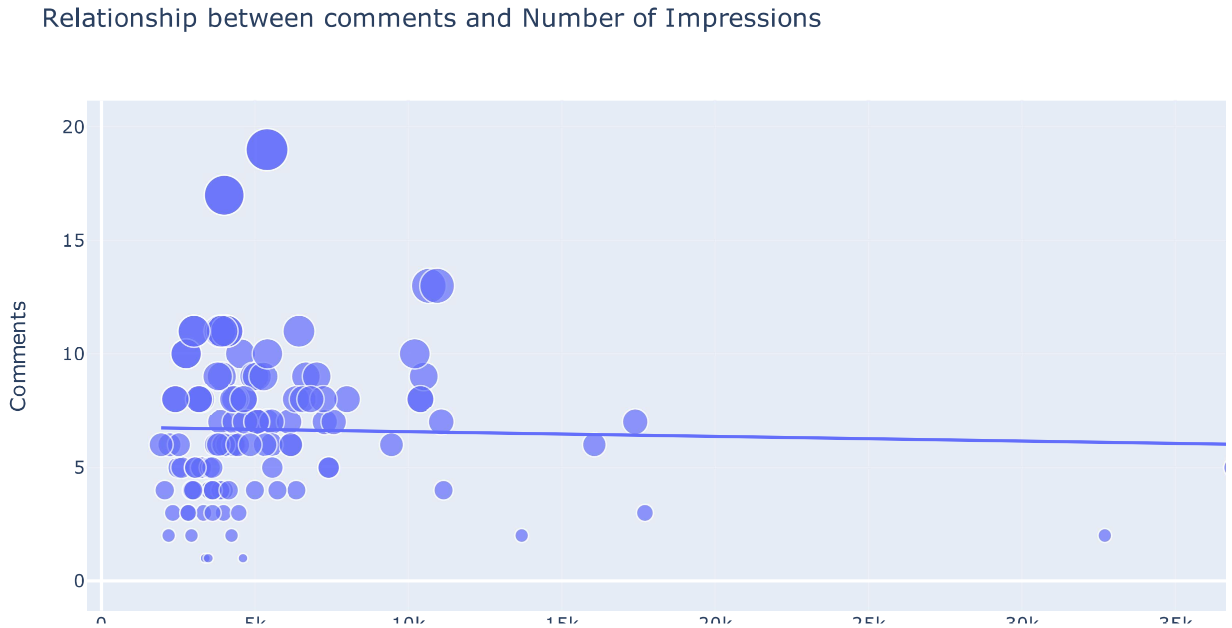
In [13]:

```
# there is a linear relationship between number of likes and impression on instagram
```

lets have a look at the relationship between number of likes and the number of impressions in instagram posts

In [14]:

```
figure = px.scatter( data_frame = df, x = "Impressions" , y = "Comments" ,trendline = "ols", size = "Comments",
                    title = "Relationship between comments and Number of Impressions" )
figure.show()
```



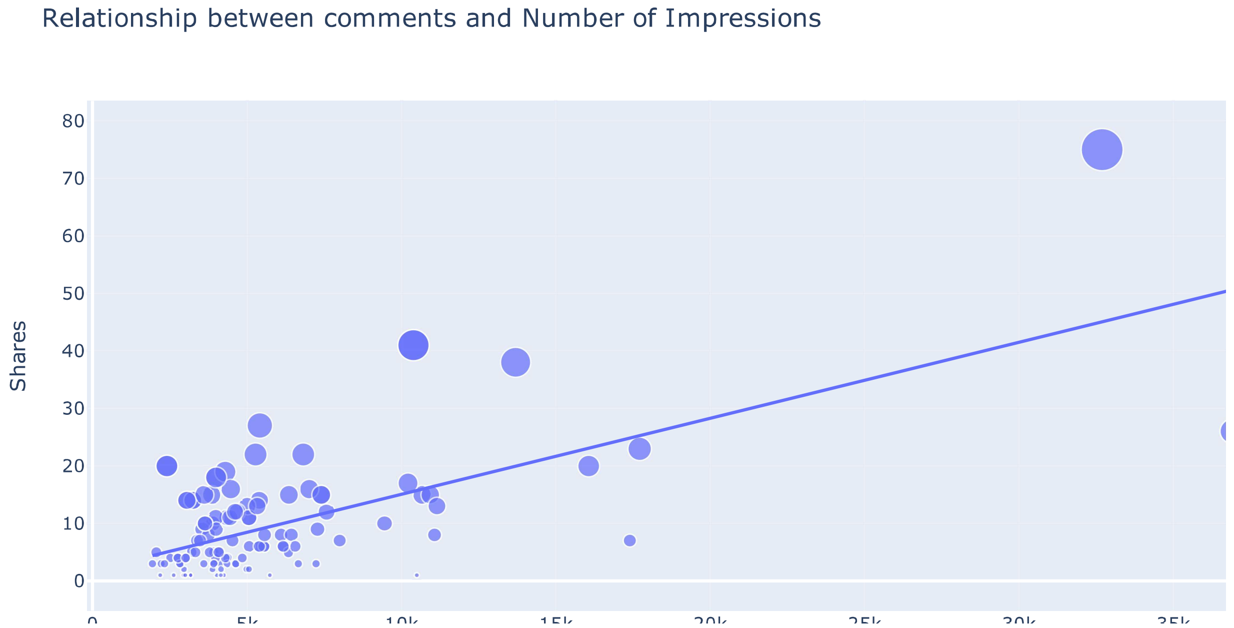
In [15]:

```
# it looks like the number of comments we get in a post does not affect its reach.
```

lets have a look at the relationship between number of shares and the number of impressions in instagram posts

In [16]:

```
figure = px.scatter( data_frame = df, x = "Impressions" , y = "Shares" ,trendline = "ols", size = "Shares",
                    title = "Relationship between comments and Number of Impressions" )
figure.show()
```



In [17]:

```
# there is a linear relationship between number of shares and impression on instagram posts. but its comparetively low 
```

correlation of all the columns with impression column

In [18]:

```
correlation = df.corr()
correlation["Impressions"].sort_values(ascending = False)
```

Out[18]:

```
Impressions        1.000000
From Explore       0.893607
Follows            0.889363
Likes              0.849835
From Home          0.844698
Saves              0.779231
Profile Visits     0.760981
Shares             0.634675
From Other         0.592960
From Hashtags      0.560760
Comments          -0.028524
Name: Impressions, dtype: float64
```

In [19]:

```
# so we can say that the higher number of likes , shared, saves will help you to get more reach.
```

Analyzing Conversion Rate

In [20]:

```
conversion rate is basically how many followers you are getting from profile visits.. The formula is( follow/profilevis
```

```
  File "C:\Users\shree\AppData\Local\Temp\ipykernel_4628\1013468296.py", line 1
    conversion rate is basically how many followers you are getting from profile visits.. The formula is(
follow/profilevisits)*100
               ^
SyntaxError: invalid syntax
```

In [ ]:

```
df.head(2)
```

In [ ]:

```
conversion_rate = (df["Follows"].sum()/df["Profile Visits"].sum())*100
```

In [ ]:

```
print("Conversion Rate is  = ", conversion_rate)
```

In [ ]:

```
# it looks good that the conversion rate is pretty high
```

In [ ]:

```
figure = px.scatter(data_frame = df, x="Profile Visits",
                    y="Follows", size="Follows", trendline="ols",
                    title = "Relationship Between Profile Visits and Followers Gained")
figure.show()
```

In [ ]:

```
# this relationship is also linear
```

In [ ]:

```
# Prediction Model
```

In [ ]:

```
x = np.array(df[["From Home",'Likes', 'Saves', 'Comments', 'Shares',
                   'Profile Visits', 'Follows',"From Hashtags","From Explore"]])
y = np.array(df["Impressions"])


X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.33, random_state=42)
```

In [ ]:

```
model = PassiveAggressiveRegressor()
model.fit(X_train, y_train)
```

In [ ]:

```
model.score( X_test, y_test)
```

In [ ]:

```

```